Computation of the Eigenelements of a Matrix by the ϵ-Algorithm

Claude Brezinski
UER d'IEEA—*Informatique*
BP *36, 59650 Villeneuve d'Ascq, France*

Recommended by N. Gastinel

ABSTRACT

In this paper we give two generalizations of the well-known power method for computing the dominant eigenvalue of a matrix. These generalizations are based on the ϵ-algorithm and allow one to compute simultaneously all the eigenvalues and the eigenvectors of a matrix. We assume here that all the eigenvalues have distinct modulus, but a forthcoming version of our algorithms will allow complex and multiple eigenvalues. Some numerical results are given which show how the method works.

1. THE POWER METHOD AND THE ϵ-ALGORITHM

The power method is a widely used method for computing the dominant eigenvalue of a square matrix. This method is often known as the Rayleigh quotient method.

Let A be a square matrix with p rows and p columns; let $\lambda_1,\dots,\lambda_p$ be its eigenvalues and $v_1,\dots,v_p$ the corresponding eigenvectors. The simplest form of the power method consists in choosing an arbitrary vector x_0 such that the scalar product (x_0,v_1) is different from zero and constructing the sequence of vectors $\{x_n\}$ by means of the iterations $x_{n+1}=Ax_n$. Setting

$$a_0^{(n)}=\frac{(y,x_{n+1})}{(y,x_n)}\qquad\text{where}\quad (y,v_1)\neq 0,$$

it can easily be seen that if

$$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_p|\geqslant 0,$$

then

$$\lim_{n\to\infty} a_0^{(n)} = \lambda_1 \quad \text{and} \quad \lim_{n\to\infty} \frac{x_n}{(y, x_n)} = v_1.$$

If there are several eigenvalues with modulus equal to $|\lambda_1|$, then the power method can be modified to converge to λ_1. In this paper we shall restrict ourselves to the first case. The interested reader is referred to the books of Faddeeva [8] and Wilkinson [14] for further considerations about this method.

The power method can also be used to compute the next eigenvalue of A. Let us assume that λ_1 has been obtained by the power method as above; then a deflation of the matrix A or a λ-difference analysis of the Rayleigh quotients allows us to compute λ_2 (see, for example [11] or [9]) if $|\lambda_2| > |\lambda_3|$. As can easily be seen, the major inconvenient of these methods is the need for a knowledge of λ_1. In other words, the eigenvalues of A can only be computed one after another: the power method must have numerically converged to λ_1 before starting new iterations for computing λ_2.

The aim of this paper is to give a generalization of the power method which allows one to compute simultaneously all the eigenvalues and the eigenvectors of A if necessary, or only some of the former if desired. This generalization is based on the ϵ-algorithm. The ϵ-algorithm is a method of accelerating the convergence of sequences of numbers due to Wynn [15]. It is a recursive process to compute Shanks' transformation of sequences [13], which is a generalized version of the well-known Δ^2 process of Aitken [1]. Wynn has also proposed a vector ϵ-algorithm working with sequences of p-dimensional complex vectors [16]. Let $\{x_n\}$ be a sequence of such vectors; the vector ϵ-algorithm consists in setting

$$\epsilon_{-1}^{(n)} = 0 \in \mathbf{C}^p, \qquad \epsilon_0^{(n)} = x_n, \qquad n = 0, 1, \ldots,$$

and then constructing the ϵ-array by use of

$$\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n+1)} + \left[\Delta\epsilon_k^{(n)}\right]^{-1} \qquad n, k = 0, 1, \ldots, \tag{1}$$

where $\Delta\epsilon_k^{(n)} = \epsilon_k^{(n+1)} - \epsilon_k^{(n)}$ and where the inverse y^{-1} of $y \in \mathbf{C}^p$ is defined as $y^{-1} = \bar{y}/(y, y)$.

This vector ϵ-algorithm has received application for solving systems of linear and nonlinear equations; as shown by Gekeler [10] and Brezinski [2,3], it produces a quadratically convergent method for systems of nonlinear equations without calculating any derivatives and under quite generous conditions. This method has been applied with success to multipoint boundary value problems for systems of ordinary differential equations by Brezinski and Rieu [7].

The use of the vector ϵ-algorithm provides a direct method for solving systems of linear equations [10], even when the matrix is singular and the system possesses infinitely many solutions [4]. Thus, as with the other direct methods for systems of linear equations, it is not surprising that the vector ϵ-algorithm gives rise to a method for computing all the eigenvalues of a square matrix.

The abstract theory of the vector ϵ-algorithm has been studied by Wynn [17], and the fundamental theorem on which the whole theory is constructed has been conjectured by Wynn [18] and proved by McLeod [12].

The assumptions and notation used in this paper are the following: A is the square $p\times p$ real matrix whose eigenelements are to be computed. $\lambda_1,\dots,\lambda_p$ are the eigenvalues of A, and $v_1,\dots,v_p$ the corresponding eigenvectors. Our hypotheses are:

$$\lambda_i \neq 1, \qquad i=1,\dots,p, \tag{2}$$

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_p| \geqslant 0 \tag{3}$$

If the first assumption (2) is not satisfied, then the matrix A can be replaced by the matrix aA, where a is a real nonzero number. Thus instead of $\lambda_1,\dots,\lambda_p$, the values $a\lambda_1,\dots,a\lambda_p$ will be obtained. The second condition (3) obviates the case where A has multiple or complex eigenvalues. This case will be treated in a forthcoming paper by a suitable modification of the algorithms studied herein. The condition (3) implies that the eigenvectors $v_1,\dots,v_p$ form a basis of $\mathbf{R}^p$.

Let $\{u_n\}$ and $\{w_n\}$ be two sequences of vectors of $\mathbf{R}^p$. The notation $u_n \sim w_n$ will be used to mean that

$$\lim_{n\to\infty} \frac{(y,u_n)}{(y,w_n)} = 1 \qquad y\neq 0 \in \mathbf{R}^p \quad \text{such that} \quad (y,v_i)\neq 0 \quad \text{for} \quad i=1,\dots,p,$$

where (y,z) denotes the scalar product between the vectors y and z. If $\{u_n\}$ and $\{w_n\}$ are sequences of real numbers, the same notation means that $\lim_{n\to\infty} u_n/w_n = 1$.

2. THE FIRST ALGORITHM

Let x_0 be an arbitrary real vector of $\mathbf{R}^p$ such that $(x_0,v_i)\neq 0\ \forall i$, and let us construct the sequence of vectors $\{x_n\}$ by $x_{n+1}=Ax_n$ for $n=0,1,\dots$. We shall now study the application of the vector ϵ-algorithm (1) to this sequence $\{x_n\}$. The main result is

THEOREM 1

$$\epsilon_{2k}^{(n)} \sim \sum_{i=k+1}^{p} \lambda_i^{n+k} z_i, \qquad k=0,\ldots,p-1,$$

$$\epsilon_{2k+1}^{(n)} \sim \frac{1}{\lambda_{k+1}^{n+k}} \frac{y_{k+1}}{\| y_{k+1}\|^2}, \qquad k=0,\ldots,p-1,$$

where $y_i = (\lambda_i - 1)z_i$ *and* $\| y_i \|^2 = (y_i, y_i)$.

Proof. Since the eigenvectors form a basis,

$$x_0 = a_1 v_1 + \cdots + a_p v_p.$$

It can, of course, be assumed that this basis is an orthogonal one, so that the condition $(x_0, v_i) \neq 0 \ \forall i$ implies $a_i \neq 0$ for $i=1,\ldots,p$. Setting $z_i = a_i v_i$, we get

$$\epsilon_0^{(n)} = x_n = \sum_{i=1}^{p} \lambda_i^n z_i,$$

$$\Delta\epsilon_0^{(n)} = \sum_{i=1}^{p} \lambda_i^n y_i,$$

$$\Delta\epsilon_0^{(n)} = \lambda_1^n \left(y_1 + \sum_{i=2}^{p} \left(\frac{\lambda_i}{\lambda_1} \right)^n y_i \right),$$

$$\|\Delta\epsilon_0^{(n)}\|^2 = \lambda_1^{2n} \left\| y_1 + \sum_{i=2}^{p} \left(\frac{\lambda_i}{\lambda_1} \right)^n y_i \right\|^2,$$

and thus we obtain

$$\epsilon_1^{(n)} \sim \frac{1}{\lambda_1^n} \frac{y_1}{\| y_1 \|^2},$$

since $y_i \neq 0 \ \forall i$ and $|\lambda_i / \lambda_1| < 1$ for $i=2,\ldots,p$.

Let us assume that the theorem has been proved until k. We have now to prove it for $k+1$:

$$\Delta\epsilon_{2k+1}^{(n)} \sim \frac{1-\lambda_{k+1}}{\lambda_{k+1}^{n+k+1}} \frac{y_{k+1}}{\| y_{k+1}\|^2},$$

$$\left[\Delta\epsilon_{2k+1}^{(n)}\right]^{-1} \sim \frac{\lambda_{k+1}^{n+k+1}}{1-\lambda_{k+1}} y_{k+1} = -\lambda_{k+1}^{n+k+1} z_{k+1},$$

by definition of y_{k+1}. We thus immediately have

$$\epsilon_{2k+2}^{(n)} \sim \sum_{i=k+1}^{p} \lambda_i^{n+k+1} z_i - \lambda_{k+1}^{n+k+1} z_{k+1} \sim \sum_{i=k+2}^{p} \lambda_i^{n+k+1} z_i,$$

which proves the theorem for the vectors $\epsilon_{2k}^{(n)}$. For the vectors with an odd lower index we get

$$\Delta\epsilon_{2k+2}^{(n)} \sim \sum_{i=k+2}^{p} \lambda_i^{n+k+1} y_i = \lambda_{k+2}^{n+k+1}\left(y_{k+2} + \sum_{i=k+3}^{p} \left(\frac{\lambda_i}{\lambda_{k+2}}\right)^{n+k+1} y_i\right)$$

and

$$\Delta\epsilon_{2k+2}^{(n)} \sim \lambda_{k+2}^{n+k+1} y_{k+2},$$

since $|\lambda_i/\lambda_{k+2}| < 1$ for $i = k+3, \ldots, p$.

$$\epsilon_{2k+3}^{(n)} \sim \frac{1}{\lambda_{k+1}^{n+k+1}} \frac{y_{k+1}}{\| y_{k+1}\|^2} + \frac{1}{\lambda_{k+2}^{n+k+1}} \frac{y_{k+2}}{\| y_{k+2}\|^2},$$

$$\epsilon_{2k+3}^{(n)} \sim \frac{1}{\lambda_{k+2}^{n+k+1}} \left(\frac{y_{k+2}}{\| y_{k+2}\|^2} + \left(\frac{\lambda_{k+2}}{\lambda_{k+1}}\right)^{n+k+1} \frac{y_{k+1}}{\| y_{k+1}\|^2}\right),$$

$$\epsilon_{2k+3}^{(n)} \sim \frac{1}{\lambda_{k+2}^{n+k+1}} \frac{y_{k+2}}{\| y_{k+2}\|^2},$$

which ends the proof of this theorem.

Our first algorithm for computing the eigenvalues of the matrix A is now clear:

(1) Let us choose an arbitrary vector x_0 such that $(x_0, v_i) \neq 0$ for $i = 1, \ldots, p$.

(2) Let us perform the iterations $x_{n+1} = Ax_n$ for $n = 0, 1, \ldots$.

(3) Let us apply the vector ϵ-algorithm (1) to the sequence $\{x_n\}$ of p-dimensional real vectors to obtain the vectors denoted by $\epsilon_k^{(n)}$.

(4) Let us form the ratios

$$a_k^{(n)} = \frac{\left(y, \epsilon_{2k}^{(n+1)}\right)}{\left(y, \epsilon_{2k}^{(n)}\right)}, \qquad k = 0, \ldots, p-1 \text{ and } n = 0, 1, \ldots,$$

or the ratios:

$$b_k^{(n)} = \frac{\left(y, \epsilon_{2k+1}^{(n)}\right)}{\left(y, \epsilon_{2k+1}^{(n+1)}\right)}, \qquad k=0,\ldots,p-1 \text{ and } n=0,1,\ldots,$$

where y is an arbitrary nonzero vector which can depend upon n and k and which can also be different for the sequences $\{a_k^{(n)}\}$ and $\{b_k^{(n)}\}$.

Using the result of Theorem 1, we immediately get

THEOREM 2. $\lim_{n\to\infty} a_k^{(n)} = \lim_{n\to\infty} b_k^{(n)} = \lambda_{k+1}$ *for* $k=0,\ldots,p-1$. *Moreover,*

$$a_k^{(n)} = \lambda_{k+1} + O\left(\left(\frac{\lambda_{k+2}}{\lambda_{k+1}}\right)^{n+k+1}\right).$$

Proof. It is evident that $b_k^{(n)} \sim \lambda_{k+1}$. For the sequence $\{a_k^{(n)}\}$ we get

$$a_k^{(n)} \sim \lambda_{k+1} \frac{(y, z_{k+1}) + \sum_{i=k+2}^{p} \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{n+k+1} (y, z_i)}{(y, z_{k+1}) + \sum_{i=k+2}^{p} \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{n+k} (y, z_i)} \sim \lambda_{k+1}.$$

The proof of the theorem can easily be achieved by effecting the division. ∎

The above algorithm can also be used to compute the eigenvectors of the matrix A:

THEOREM 3.

$$\lim_{n\to\infty} \frac{\epsilon_{2k}^{(n)}}{\left(y, \epsilon_{2k}^{(n)}\right)} = \lim_{n\to\infty} \left(y, \epsilon_{2k+1}^{(n)}\right) \epsilon_{2k}^{(n)} = v_{k+1} \qquad \text{for } k=0,\ldots,p-1.$$

Proof. By Theorem 1 we get

$$\epsilon_{2k}^{(n+1)} \sim \sum_{i=k+1}^{p} a_i \lambda_i^{n+k+1} v_i = A \sum_{i=k+1}^{p} a_i \lambda_i^{n+k} v_i = A \epsilon_{2k}^{(n)}.$$

Dividing both members by $(y, \epsilon_{2k}^{(n)})$, the preceeding relation can be written as

$$\frac{\left(y, \epsilon_{2k}^{(n+1)}\right)}{\left(y, \epsilon_{2k}^{(n)}\right)} \frac{\epsilon_{2k}^{(n+1)}}{\left(y, \epsilon_{2k}^{(n+1)}\right)} \sim A \frac{\epsilon_{2k}^{(n)}}{\left(y, \epsilon_{2k}^{(n)}\right)}.$$

This relation proves the first part of this theorem by using the first convergence result of Theorem 2.

Multiplying both members of the same relation by $(y, \epsilon_{2k+1}^{(n)})$, we get

$$\frac{\left(y, \epsilon_{2k+1}^{(n)}\right)}{\left(y, \epsilon_{2k+1}^{(n+1)}\right)} \left(y, \epsilon_{2k+1}^{(n+1)}\right) \epsilon_{2k}^{(n+1)} \sim A \left(y, \epsilon_{2k+1}^{(n)}\right) \epsilon_{2k}^{(n)},$$

which ends the proof by using the second convergence result of Theorem 2. ■

3. THE SECOND ALGORITHM

Instead of the vector ϵ-algorithm, the repeated application of the vector form of the Aitken Δ^2-process can be used. By repeated application we mean the following method: Let us first apply the vector ϵ-algorithm to a sequence of vectors $\{x_n\}$ to produce the vectors denoted by $\epsilon_2^{(n)}$. This sequence $\{\epsilon_2^{(n)}\}$ can now be used as an initial sequence for a second application of the vector ϵ-algorithm, and so on. This method can be formalized by the notation

$${}_0\epsilon_0^{(n)} = x_n, \qquad n = 0, 1, \ldots .$$

For the $(k+1)$th repeated application $(k = 0, 1, \ldots)$,

$${}_k\epsilon_1^{(n)} = \left[\Delta_k \epsilon_0^{(n)}\right]^{-1}$$

$${}_k\epsilon_2^{(n)} = {}_k\epsilon_0^{(n+1)} + \left[\Delta_k \epsilon_1^{(n)}\right]^{-1}$$

$${}_{k+1}\epsilon_0^{(n)} = {}_k\epsilon_2^{(n)} \qquad \text{for } n = 0, 1, \ldots . \tag{4}$$

If $\{x_n\}$ is a sequence of vectors produced by $x_{n+1} = Ax_n$, where x_0 is an arbitrary vector such that $(x_0, v_i) \neq 0$ for $i = 1, \ldots, p$, then we obtain

THEOREM 4.

$$ {}_k\epsilon_0^{(n)} \sim \sum_{i=k+1}^{p} \lambda_i^{n+k} z_i, \qquad k=0,\ldots,p-1, $$

$$ {}_k\epsilon_1^{(n)} \sim \frac{1}{\lambda_{k+1}^{n+k}} \frac{y_{k+1}}{\| y_{k+1} \|^2}, \qquad k=0,\ldots,p-1. $$

Proof.

$$ {}_0\epsilon_0^{(n)} = x_n = \sum_{i=1}^{p} \lambda_i^n z_i. $$

Let us assume that the theorem has been proved until k, and let us prove it for $k+1$:

$$ \Delta\, {}_k\epsilon_0^{(n)} \sim \sum_{i=k+1}^{p} \lambda_i^{n+k} y_i = \lambda_{k+1}^{n+k} \left(y_{k+1} + \sum_{i=k+2}^{p} \left(\frac{\lambda_i}{\lambda_{k+1}} \right)^{n+k} y_i \right), $$

$$ \Delta\, {}_k\epsilon_0^{(n)} \sim \lambda_{k+1}^{n+k} y_{k+1}. $$

Thus we get

$$ {}_k\epsilon_1^{(n)} \sim \frac{1}{\lambda_{k+1}^{n+k}} \frac{y_{k+1}}{\| y_{k+1} \|^2} \qquad \Delta\, {}_k\epsilon_1^{(n)} \sim \frac{1-\lambda_{k+1}}{\lambda_{k+1}^{n+k+1}} \frac{y_{k+1}}{\| y_{k+1} \|^2} $$

and

$$ \left[\Delta\, {}_k\epsilon_1^{(n)} \right]^{-1} \sim -\lambda_{k+1}^{n+k+1} z_{k+1} \qquad \text{(by definition of } y_{k+1}\text{)}, $$

$$ {}_k\epsilon_2^{(n)} \sim \lambda_{k+1}^{n+k+1} z_{k+1} + \sum_{i=k+2}^{p} \lambda_i^{n+k+1} z_i - \lambda_{k+1}^{n+k+1} z_{k+1}, $$

which ends the proof of the theorem, since ${}_{k+1}\epsilon_0^{(n)} = {}_k\epsilon_2^{(n)}$.

Thus our second algorithm for computing the eigenvalves of A is the following:

(1) Let us choose an arbitrary vector x_0 such that $(x_0, v_i) \neq 0$ for $i=1,\ldots,p$.

(2) Let us perform the iterations $x_{n+1} = Ax_n$ for $n=0,1,\ldots$.

(3) Let us make the repeated application (4) to the sequence $\{x_n\}$ of p-dimensional real vectors to obtain the vectors denoted by ${}_k\epsilon_0^{(n)}$ and ${}_k\epsilon_1^{(n)}$.

(4) Let us form the ratios:

$$ c_k^{(n)} = \frac{\left(y, {}_k\epsilon_0^{(n+1)} \right)}{\left(y, {}_k\epsilon_0^{(n)} \right)}, \qquad k=0,\ldots,p-1 \text{ and } n=0,1,\ldots, $$

or the ratios:

$$d_k^{(n)} = \frac{\left(y, {}_k\epsilon_1^{(n)}\right)}{\left(y, {}_k\epsilon_1^{(n+1)}\right)}, \qquad k=0,\dots,p-1 \text{ and } n=0,1,\dots,$$

where y is an arbitrary nonzero vector which can depend upon n and k and which can also be different for the sequences $\{c_k^{(n)}\}$ and $\{d_k^{(n)}\}$.

THEOREM 5. $\lim_{n\to\infty} c_k^{(n)} = \lim_{n\to\infty} d_k^{(n)} = \lambda_{k+1}$ *for* $k=0,\dots,p-1$. *Moreover,*

$$c_k^{(n)} = \lambda_{k+1} + O\left(\left(\frac{\lambda_{k+2}}{\lambda_{k+1}}\right)^{n+k+1}\right).$$

THEOREM 6.

$$\lim_{n\to\infty} \frac{{}_k\epsilon_0^{(n)}}{\left(y, {}_k\epsilon_0^{(n)}\right)} = \lim_{n\to\infty} \left(y, {}_k\epsilon_1^{(n)}\right) {}_k\epsilon_0^{(n)} = v_{k+1} \qquad \text{for } k=0,\dots,p-1.$$

The proofs of Theorems 5 and 6 are exactly the same that the proofs of Theorems 2 and 3 because the result of Theorem 4 is identical to the result of Theorem 1.

It is evident that the two preceding methods can be used to compute only some of the dominant eigenvalues, since they are found in decreasing order.

4. VARIANTS OF THE ALGORITHMS

In the rules for the vector ϵ-algorithm (1) or in the rules for its repeated application (4), the inverse y^{-1} of a vector $y \in \mathbf{R}^p$ can be defined as

$$y^{-1} = y/\|y\|^2,$$

where $\|\ \|$ denotes any vectorial norm instead of the Euclidian norm. We thus obtain a particular case of an ϵ-algorithm proposed by Brezinski [5] to accelerate the convergence of sequences in a Banach space.

In the numerical applications, certain other norms are easier to compute than the Euclidian norm and are less subject to roundoff errors. For our purpose these variants of our algorithms have the same properties, and Theorems 1 and 6 still remain valid. There is nothing else to prove, as the proofs have been given with a norm which can be chosen arbitrarily.

Instead of (1) and (4) we can also use a new ϵ-algorithm proposed by Brezinski [6] to accelerate the convergence of sequences in a topological vector space. We don't want to develop here any theoretical considerations about this algorithm; let us only say that, in the vector case, Theorems 1 and 6 are still true, and that this algorithm can be used to compute the eigenvalue of some linear operators in topological vector spaces. This subject will be studied further.

Let us now speak about a modification of our methods which considerably reduces the computational effort involved in the algorithms but does not provide us the eigenvectors of the matrix. Instead of applying the vector ϵ-algorithm or the vector form of the Aitken process to the complete vectors $\{x_n\}$, this modification consists in applying them to vectors formed only by some components of the vectors $\{x_n\}$.

If the vector x_0 still satisfies $(x_0, v_i) \neq 0$ for $i = 1, \ldots, p$, then the same theorems are still true for the reduced vectors $\epsilon_k^{(n)}$. Thus the eigenvalues can be computed using only some components of the vectors $\{x_n\}$. This modification reduces the computations and also the storage needed for the algorithms, but provides us only some of the components of the eigenvectors. In particular, our methods can be applied to a single component of the x_n. In this case our algorithms are nothing else but the scalar ϵ-algorithm or the repeated Δ^2 process applied to the sequence $\{(x_n)_k\}$ for $n = 0, 1, \ldots,$ where $(x_n)_k$ designs the kth component of x_n. We thus obtain the eigenvalues of A and the kth components of all the eigenvectors. If this variant is used for every $k = 1, \ldots, p$, we get all the components of the eigenvectors; this can be done with the vector forms of our algorithms, where the inverse of a vector y is y^{-1} such that $(y^{-1})_k = 1/y_k$.

5. ACCELERATION OF THE CONVERGENCE

It is well known that the scalar Δ^2 process of Aitken accelerates the convergence of the sequence $\{a_0^{(n)}\}$ produced by the power method. It can be shown that if $a_0^{(n)} = \lambda_1 + O\left((\lambda_2/\lambda_1)^{n+1}\right)$, then we get

$$\epsilon_2^{(n)} = \frac{a_0^{(n+2)} a_0^{(n)} - \left[a_0^{(n+1)}\right]^2}{a_0^{(n+2)} - 2a_0^{(n+1)} + a_0^{(n)}} = \lambda_1 + O\left(\left(\frac{\lambda_3}{\lambda_1}\right)^{n+1}\right).$$

The use of the complete scalar ϵ-algorithm as defined by Wynn [15] is a very powerful device to acceletate the convergence of sequences such as $\{a_k^{(n)}\}$, and Wynn [19] proved that if the ϵ-algorithm is applied to the sequence

$$\epsilon_0^{(n)} \sim a + \sum_{i=1}^{\infty} a_i b_i^n \qquad \text{with } 1 > b_1 > b_2 > \cdots > 0,$$

then for fixed k, $\epsilon_{2k}^{(n)} \sim a + O(b_{k+1}^n)$. The same result holds if $-1 < b_1 < b_2 < \cdots < 0$. The result is also true whatever be the constants b_i for $i = 1, 2, \ldots$; this is nothing but Theorem 1 for scalars and $k = 0, 1, \ldots$.

We thus get immediately

THEOREM 7. *If the scalar ϵ-algorithm is applied to the sequence $\{a_k^{(n)}\}$ for fixed k or to the sequence $\{c_k^{(n)}\}$ for fixed k, then it produces numbers denoted by $\epsilon_{2q}^{(n)}$ which satisfy the relation*

$$\epsilon_{2q}^{(n)} \sim \lambda_{k+1} + O\left(\left(\frac{\lambda_{k+q+2}}{\lambda_{k+1}}\right)^{n+k}\right)$$

for fixed q and $k = 0, \ldots, p-1$.

Proof. From the proof of Theorem 2 it is easy to derive that

$$a_k^{(n)} \sim \lambda_{k+1} + \sum_{i=k+2}^{p} \alpha_i \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{n+k} + e_{k,n},$$

where α_i are nonzero coefficients and where $e_{k,n}$ designates a term which is infinitely small before the summation. The end of the proof follows from the preceeding result about the convergence of the scalar ϵ-array. ■

The consequence of this theorem is that the sequence $\{\epsilon_{2q+2}^{(n)}\}$ converges to λ_{k+1} faster than the sequence $\{\epsilon_{2q}^{(n)}\}$ for fixed q, in the sense that

$$\lim_{n\to\infty} \frac{\epsilon_{2q+2}^{(n)} - \lambda_{k+1}}{\epsilon_{2q}^{(n)} - \lambda_{k+1}} = 0,$$

when the scalar ϵ-algorithm is applied to $\epsilon_0^{(n)} = a_k^{(n)}$ or to $\epsilon_0^{(n)} = c_k^{(n)}$, $n = 0, 1, \ldots$.

6. NUMERICAL EXAMPLE

Let us now consider the following matrix:

$$\begin{pmatrix} 3 & 12 & 30 \\ -6 & -27 & -66 \\ 4 & 16 & 37 \end{pmatrix},$$

whose eigenelements are

$$\lambda_1=9, \qquad \lambda_2=3, \qquad \lambda_3=1,$$

$$v_1=\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}, \qquad v_2=\begin{pmatrix} 3 \\ -5 \\ 2 \end{pmatrix}, \qquad v_3=\begin{pmatrix} 3 \\ -3 \\ 1 \end{pmatrix}.$$

Let us start our first algorithm with $x_0=(1;0;0)^T$, and let us choose $y=(1;0;0)^T$; we get

$\{a_0^{(n)}\}$	$\{b_0^{(n)}\}$	$\{a_1^{(n)}\}$	$\{b_1^{(n)}\}$	$\{a_2^{(n)}\}$	$\{b_2^{(n)}\}$
3.					
19.	12.9				
11.4	10.1	−2.5			
9.7	9.3	5.2	2.93		
9.2	9.1	3.4	2.98	0.999999	
9.07	9.04	3.1	2.992	1.000000	8.5
9.02	9.01	3.04	2.997	0.999999	5.3
9.008	9.004	3.01	2.9991	1.000000	1.7
9.002	9.001	3.004	2.9997	0.999999	1.05
9.0009	9.0004	3.001	2.99991	1.000000	0.996
9.0003	9.0001	3.0004	2.99997	1.000000	0.996
9.0001	9.00005	3.0001	2.99999	0.999999	0.998

As stated by Theorem 7, the convergence of these sequences can be accelerated by using the scalar ϵ-algorithm. Let us, for example, consider the sequences $\{b_1^{(n)}\}$; we get

$\{\epsilon_0^{(n)} = b_1^{(n)}\}$	$\{\epsilon_2^{(n)}\}$	$\{\epsilon_4^{(n)}\}$	$\{\epsilon_6^{(n)}\}$	$\{\epsilon_8^{(n)}\}$
2.93				
2.98	2.99992			
2.992	2.999991	2.999999989		
2.997	2.9999990	2.9999999996	3.0000000002	
2.9991	2.99999989	3.0000000001	2.9999999998	2.999999999998
2.9997	2.999999988	2.9999999996	2.99999999991	
2.99991	2.9999999983	3.0000000004		
2.99997	3.00000000008			
2.99999				

For the eigenvectors we obtain

with $\{\epsilon_0^{(n)}\}$, $(1;\ -2.000000002;\ 1.000000002)^T$
with $\{\epsilon_1^{(n)}\}$, $(0.1874999992;\ -0.3749999990;\ 0.1874999997)^T$
with $\{\epsilon_2^{(n)}\}$, $(1;\ -1.66666667;\ 0.66666667)^T$
with $\{\epsilon_3^{(n)}\}$, $(0.355263157;\ -0.592105264;\ 0.23682106)^T$
with $\{\epsilon_4^{(n)}\}$, $(1;\ -0.99999999;\ 0.33333333)^T$
with $\{\epsilon_5^{(n)}\}$, $(-0.53421011;\ 0.53421011;\ -0.17807003)^T$

I should like to thank M. F. Rachdi for the computation of the numerical example. It is also a pleasure to thank Prof. P. Wynn for helpful conversations about this paper and other matters.

REFERENCES

1 A. C. Aitken, On Bernouilli's numerical solution of algebraic equations, *Proc. Roy. Soc. Edinburgh* **46** (1926), 289–305.
2 C. Brezinski, Application de l' ϵ-algorithme à la résolution des systèmes non linéaires, *C. R. Acad. Sci. Paris* **271A** (1970), 1174–1177.
3 C. Brezinski, Sur un algorithme de résolution des systèmes non linéaires, *C. R. Acad. Sci. Paris*, **272A** (1971), 145–148.
4 C. Brezinski, Some results in the theory of the vector ϵ-algorithm, *Linear Algebra* **8** (1974), 77–86.
5 C. Brezinski, Accélération de la convergence de suites dans un espace de Banach, *C. R. Acad. Sci. Paris* **278A** (1974), 351–354.
6 C. Brezinski, Généralisations de la transformation de Shanks, de la table de Padé et de l' ϵ-algorithme, to appear in *Calcolo.*
7 C. Brezinski and A. C. Rieu, The solution of systems of equations using the ϵ-algorithm and an application to boundary value problems, *Math. Comp.* **28** (1974), 731–741.
8 V. N. Faddeeva, *Computational methods of linear algebra,* Dover, New York, 1959.

9 N. Gastinel, *Analyse numerique linéaire*, Hermann, Paris, 1966.

10 E. Gekeler, On the solution of systems of equations by the epsilon algorithm of Wynn, *Math. Comp.* **26** (1972), 427–436.

11 E. Isaacson and H. B. Keller, *Analysis of numerical methods*, Wiley, New York, 1966.

12 J. B McLeod, A note on the ϵ-algorithm, *Computing* **7** (1971) 17–24.

13 D. Shanks, Nonlinear transformations of divergent and slowly convergent series, *J. Math. Phys.* **34** (1955), 1–42.

14 J. H. Wilkinson, *The algebraic eigenvalue problem*, Clarendon Press, Oxford, 1965.

15 P. Wynn, On a device for computing the $e_m(S_n)$ transformation, *MTAC* **10** (1956), 91–96.

16 P. Wynn, Acceleration techniques for iterated vector and matrix problems, *Math. Comp.* **16** (1962), 301–322.

17 P. Wynn, Vector continued fractions, *Linear Algebra* **1** (1968), 357–395.

18 P. Wynn, Upon a conjecture concerning a method for solving linear equations and certian other matters, MRC Technical Summary Report 626, 1966.

19 P. Wynn, On the convergence and stability of the epsilon algorithm, *SIAM J. Numer. Anal.* **3** (1966), 91–122.

Received 27 May 1974